Student Name: **Student id:** **Sect#:** **Serial #:**

QUESTION #	1	2	3	4	TOTAL
MAX POINTS	16	17	17	18	68
POINTS EARNED					

University of Bahrain College of Information Technology
Department of Computer Science

ITCS332: Organization of Programming Languages **SECOND TEST** **Date: DEC 7, 2015**

****

QUESTION ONE: **[7+9 pts]**

a) **What will be printed after executing the following C++ code?**

```
1) static int t=9;
2) void funX()
3) { static int x = 4;
4)   int f = x * t;
5)   int *p = new int(8);
6)   x = x + t - *p;
7)   t++; *p = *p + 6;
8)   cout << x << '\t' << f << '\n';
9)   delete p;
10)  }
11)  void main()
12)  {
13)     cout << "t = " << t << '\t' ;
14)     funX();
15)     cout << "t = " << t << '\t' ;
16)     funX();
17)  }
```

```
t = 9    5        36
t = 10   7        50
```

b) **Using the above given C++ code, fill in blanks as required**

1) The lifetime of a variable ***f*** begins whenever ***the function funX is called*** and ends whenever ***the function funX is terminated (return).***
2) The type of the variable ***f*** is ***stack-dynamic***
3) The type of the variable (object) pointed to by **p** is ***explicit-heap dynamic***
4) The scope of the pointer variable **p** is ***from line #5 till line #10.***
5) The scope of the variable **t** is ***from line #1 till line #17.***
6) The variable pointed to by ***p*** in bound to storage whenever ***the new operator in line #5 is executed*** and unbound whenever ***the delete statement in line#9 is executed.***
7) The type is bound to a variable ***f*** during the ***compilation*** time. The type is bound to a variable pointed to by ***p*** during the ***compilation*** time.
8) The variable ***f*** is allocated space on the ***stack*** and the variable pointed by ***p*** is allocated space on the ***heap***.
9) The lifetime of a variable **x** begins when **the function *funX* is loaded first time** and ends when **the entire *program* terminates.**

QUESTION TWO: **[8+9 pts]**

a) **Consider the following Ada-like code.** The values of n and m printed by **print(n,m):**

```
Procedure main is
  n,m: integer ;
  Procedure sub1 is
    begin
      n = n - 8 ; m *= 2; print(n,m);
    end ;
  Procedure sub2 is
    n : integer;
    begin
       n = 18 ; m = n - 7;sub1;
    end ;
begin
    n = 20 ; m = 23; sub2;
end ;
```

1) Under static -scoped rules are: ***n=20-8=12***
   m=23*2=46

2) Under dynamic-scoped rules are: ***n=18-8=10***
   m=11*2=22

b) **Fill in blanks as required**

1) A data type is a **collection of data values (objects)** plus **a set of predefined operations that can be applied on those objects.**
2) The dynamic length strings are implemented using **Linked Lists** or **Adjacent memory Cells.**
3) Name two character string design issues: **string type: primitive or structured** and **string length: static or dynamic.**
4) The two major disadvantages of static variables are: **They do not support recursion** and **Low efficient use of memory space.**
5) The main advantage of coercion is its **flexibility** (**better writeability**), and its disadvantage is **less reliability** (**some type errors cannot be detected)**.
6) Data type descriptors are used by the compiler for **type checking** and **to build the code for allocation and deallocation operations**
7) The precision of the floating point value depends **on the length of the fraction** in its representation. The range of the floating point value depends **on the exponent and fraction lengths**.
8) In packed BCD, the decimal number 79 is represented as **0111 1001** and in binary as **0100 1111**.
9) Using 2's complement to represent signed integer values is decided by **language implementer**. The number of bytes allocated to each primary type is decided by **language implementer**.

QUESTION THREE: **[8+9 pts]**

Part #1

1) A Prolog statement consists of terms which may be: ***constant*** or ***variable***
2) Prolog operates in ***entry*** or ***query*** modes.
3) Name 2 kinds of Prolog statements: ***fact*** and ***rule***
4) The right side of a rule is called ***antecedent*** and the left side is called ***consequent***
5) The prolog query `?- X = 35, X is 22+13.` produces ***X = 35.***
6) The prolog query *?- [b, 34, 5|U] = [b, 34, 5, f, 22,f,isa].* produces ***U=[f,22,f,isa]***
7) The value of U produced by the query `?- X=[isa,hud],Y=[f,g,h], U=[Y|X].` is
   U =[[f, g, h], isa, hud]
8) The value of L3 produced by the query `?- G =[f|L3],L3=[ h,b,t|L2],L2=[s,f,9].` is:
   L3 = [h, b, t, s, f, 9]

Part #2

1) "*X and Y are brothers if they have the same mother and the same father*" .The Prolog rule that describes this relation is:

```
brothers(X,Y):- father(F,X),father(F,Y),mother(M,X),mother(M,Y).
```

```
myst([],0).
myst([X|Y], R):- myst(Y,U), R is U+X-2.
```

2) The prolog query *?-* `myst([5,13,11],T).` produces ***U = 23.***

3) Write Prolog predicate(s) named "***heads***" that accepts a list of lists L and produces a list consisting of the head elements in the given L as shown in the following queries.

```
?- heads([[she,7],[has,b,20],[passed,fail]],L).
L = [she,has,passed].

?- heads([[99,88],[tt,b,isa],[77,a,b],[musa]],T).
T = [99,tt,77,musa].
```

```
heads([],[]).
heads([[H|_]|T],[H|LN]):- heads(T,LN).
```

QUESTION FOUR: **[10+8 pts]**

Part #1

1) The main advantage of explicit heap-dynamic variables is efficient use of memory. Justify. **[2 pts]**

   Memory is allocated by the user when needed and immediately deallocated when not needed .

2) Give one advantage and one disadvantage of long names of variables **[2 pts]**

   The advantage: Long names are meaningful (Improve the readability).

   The disadvantage: Long names increase the size of the symbol table in memeory.

3) A variable may have multiple addresses at different places. Give C++ code that illustrates that. **[2 pts]**

```
void f1(...)
{  int x; ...  }

void f2(...)
{  int x; f1(…); ...      }
```

4) The advantage of static variables is no time overhead. Justify. **[2 pts]**

   No execution time wasted for allocation/deallocation: Static variables are allocated during the first loading of a program unit and deallocated when the program terminates.

5) The disadvantage of static variables is inefficient use of memory. Justify. **[2 pts]**

   Static variables occupy memory for very long time: from the first loading of a program unit and until the exit from the program.

Part #2 **[8 pts]**

6) In programming languages, aliases are created using **references** or **parameters**.
7) In C++, the static scope is created for every **block** and **function**.
8) In static-scoped languages, the type of a variable is specified using **implicit declaration** or **explicit declaration.**
9) In C++, the "`&&`" symbol is bound to "and" operation at **language design** time, and a call to a library function "`sqrt`" is bound to the function code in the library at **link** time.
10) In C++, a variable created with **new** operator is bound to a type during the **compilation** time and bound to a memory location during the **execution** time.
11) In static-scoped languages, the referencing environment is based on **the textual layout** of the program units/blocks. In dynamic-scoped languages, the referencing environment is based on **the calling sequence** of the program units/blocks.
12) A stack-dynamic variable is allocated space **on every entry** of the defining unit/block and deallocated **on every exit** of the defining unit/block.
13) The two disadvantages of dynamic length string implemented as a linked list are**: complex and slow string operations** and **pointer space overhead**.